Evolving Solutions to Community-Structured Satisfiability Formulas

Frank Neumann
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, Australia

Andrew M. Sutton
Department of Computer Science
University of Minnesota Duluth
Duluth, MN, USA

Abstract

We study the ability of a simple mutation-only evolutionary algorithm to solve propositional satisfiability formulas with inherent community structure. We show that the community structure translates to good fitness-distance correlation properties, which implies that the objective function provides a strong signal in the search space for evolutionary algorithms to locate a satisfying assignment efficiently. We prove that when the formula clusters into communities of size $s \in \omega(\log n) \cap O(n^{\varepsilon/(2\varepsilon+2)})$ for some constant $0 < \varepsilon < 1$, and there is a nonuniform distribution over communities, a simple evolutionary algorithm called the (1+1) EA finds a satisfying assignment in polynomial time on a $1 - o(1)$ fraction of formulas with at least constant constraint density. This is a significant improvement over recent results on uniform random formulas, on which the same algorithm has only been proven to be efficient on uniform formulas of at least logarithmic density.

Introduction

Evolutionary algorithms are a broad class of incomplete randomized search heuristics that are sometimes applied to tackling difficult optimization problems that arise in practice. We study the performance of a simple evolutionary algorithm tasked with finding a satisfying assignment to structured (non-uniform) propositional formulas expressed as the conjunction of m Boolean disjunctive clauses of size exactly k over n variables. A series of recent papers (Sutton and Neumann 2014; Doerr, Neumann, and Sutton 2017; Buzdalov and Doerr 2017) have derived rigorous running time bounds on a variety of evolutionary algorithms applied to solving propositional formulas generated uniformly at random.

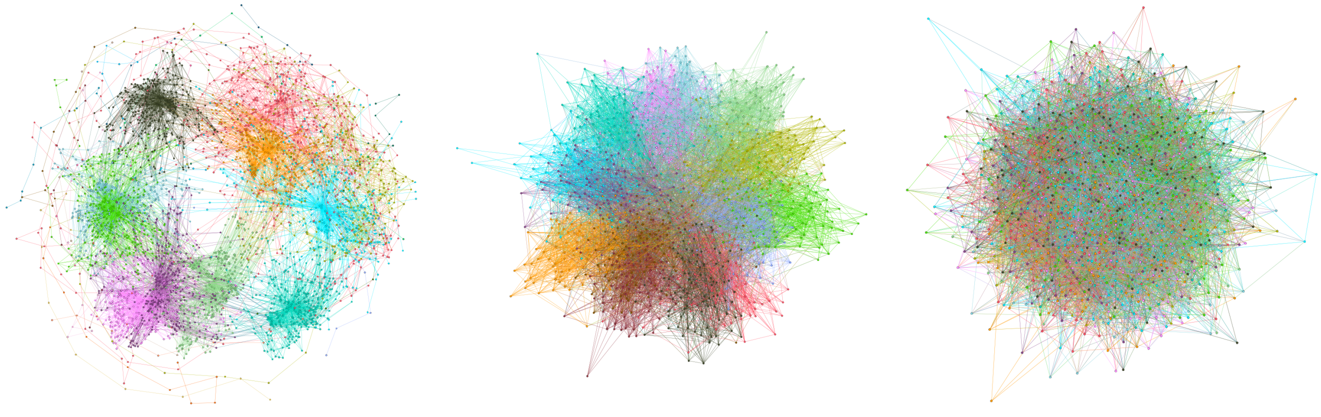
One criticism of the uniform random model is that it inadequately characterizes propositional formulas that arise from practical applications. Industrial SAT instances are far more complex, and contain complicated structural characteristics. Among these are *modularity* (Giráldez-Cru and Levy 2016), *heterogeneity* (Ansótegui, Bonet, and Levy 2009), *self-similarity* (Ansótegui et al. 2014), and *locality* (Giráldez-Cru and Levy 2017). Several non-uniform formula distributions have been proposed to model the above characteristics better.

In this paper, we are interested in extending the analysis to structured formulas that exhibit features that are statistically closer to industrial satisfiability problems. We conduct an analysis of the (1+1) EA over a generalization of the *community attachment* model of Giráldez-Cru and Levy (2016) for modular SAT formulas. Figure 1 illustrates the constraint graph structure for an industrial formula (from bounded model checking), a formula generated by the community attachment model, and a formula generated uniformly at random.

We show that the (1+1) EA can solve in polynomial time a $1 - o(1)$ fraction of satisfiable formulas drawn from the community attachment model with constraint density (average degree) that is at least a sufficiently large constant, provided that the density within communities is nonuniform. This bound covers a larger region of the constraint density spectrum than previous results on evolutionary algorithms on the uniform random model. In particular, analogous results were proved for the (1+1) EA on the uniform random model, but only for formulas of density at least $\Omega(\log n)$ (Doerr, Neumann, and Sutton 2017). For polylogarithmic densities, a more complicated evolutionary algorithm called the (1+(λ,λ)) GA yields a $\sqrt{\log n}$ speed up on formulas with density $\omega(\log^2 n)$, and even further improvements on asymptotically larger densities when the population size is adapted (Buzdalov and Doerr 2017).

Randomized search heuristics such as WalkSAT (Selman, Kautz, and Cohen 1994), Schöning's algorithm (Schöning 1999) and evolutionary algorithms are *incomplete* in the sense that they can only locate a satisfying assignment if it exists, and not decide an unsatisfiable formula. A performance guarantee on satisfiable formulas for such algorithms allows for the construction of a Monte Carlo decision algorithm with one-sided error (Sutton and Neumann 2014). It therefore suffices to derive time bounds only on satisfiable formulas.

Ideally, given an instance distribution, one would like to derive performance guarantees over the distribution conditioned on satisfiability. This is sometimes referred to as *filtering* (Kautz et al. 2001). Unfortunately, filtering is difficult in practice (as it requires a complete solver to decide satisfiability), and creates complicated dependencies



(a) Industrial formula

(b) Community attachment

(c) Uniform random

Figure 1: Constraint graphs for different formulas partitioned into community components by the Gephi tool (Bastian, Heymann, and Jacomy 2009). The industrial formula is a bounded model checking instance. The synthetic formulas both have $n = 1162$, $m = 3738$ and $k = 3$. The community attachment formula has $s = 83$ and $p = 0.696$.

that are difficult to control in analysis. To address this, previous work analyzing the running time of evolutionary algorithms on the random propositional satisfiability model (Sutton and Neumann 2014; Doerr, Neumann, and Sutton 2017; Buzdalov and Doerr 2017) investigate the fitness-distance correlation on the *random planted* model, in which formulas are forced to be satisfiable by hiding some satisfying assignment within the formula. For reasonably dense formulas, this corresponds to the uniform random filtered model, i.e., the uniform model conditioned on satisfiability (Doerr, Neumann, and Sutton 2017). We point out that random planted formulas are easy to solve for classical algorithms (Krivelevich and Vilenchik 2006), but our goal is to advance the theoretical analysis of incomplete search heuristics on random satisfiability models.

To derive performance guarantees on satisfiable community-structured formulas, we investigate a *planted modular* satisfiability model. In this model, we choose uniformly at random an assignment $x^\star$ from the set of all assignments. For each k-set of Boolean variables, we only allow clauses from the $2^k - 1$ unique clauses on those variables that are satisfied by $x^\star$ (rather than the original 2^k). Similar to the uniform model, for reasonably dense formulas the planted modular SAT model converges to the filtered modular SAT model. The proof of this claim is a straightforward adaptation of the analogous proof for the uniform model (Doerr, Neumann, and Sutton 2017, Theorem 4), and we omit it for space.

Formally, we consider propositional k-CNF formulas F over n Boolean variables:

$$F = \bigwedge_{i=1}^{m} (\ell_{i,1} \vee \ell_{i,2} \vee \cdots \vee \ell_{i,k})$$

where $\ell_{i,j}$ is one of the n variables or its negation. The set of n variables corresponds to the set $[n] := \{1, 2, \ldots, n\}$, and we will refer to these sets interchangeably. Let $s := s(n)$ be a divisor of n. The community attachment model (Giráldez-Cru and Levy 2016) is defined as the set of all propositional k-CNF formulas $\mathcal{F}(n, m, k, s, p)$ that are constructed as follows. Partition $[n]$ into $t = n/s$ disjoint subsets $\Gamma_1, \ldots, \Gamma_t$ where

$$\Gamma_\ell = \{s(\ell - 1) + 1, s(\ell - 1) + 2, \ldots, s\ell\}.$$

We call a k-CNF clause *localized* if it contains only k literals that involve variables from the same community. We call a clause *separated* if it does not contain any pairs of literals that involve variables from the same community. Each clause is generated by drawing a localized clause with probability p uniformly at random, and a separated clause uniformly at random with probability $1 - p$. Note that, except when $k = 2$, the model does not contain uniform random k-SAT as a special case. When $k = 2$, setting $p = (s - 1)/(n - 1)$ recovers the uniform random 2-SAT model. Empirically, the model correlates better to industrial formulas in the sense that solvers that specialize on industrial instances perform better on highly modular instances than on formulas that are closer to uniform (Giráldez-Cru and Levy 2016).

In real world data, the density of communities is not uniform. For example, the density of each community for the industrial bounded model checking formula shown on the left of Figure 1 is charted in Figure 2. To address this phenomenon, we generalize the community attachment model of Giráldez-Cru and Levy (2016). In particular we define a distribution $0 \leq \pi_\ell \leq 1$ for $\ell \in [t]$ such that $\sum_{\ell=1}^{t} \pi_\ell = 1$. A localized clause is chosen from community ℓ with probability π_ℓ. The original community attachment model is recovered with the uniform distribution $\pi_\ell = 1/t$ for all $\ell \in [t]$.

Let F be a propositional formula on n vertices expressed as the conjunction of m disjunctive clauses. The *constraint graph* of F is an undirected graph $G = (V, E)$ whose n vertices V represent the variables of F, and $(u, v) \in E$ if

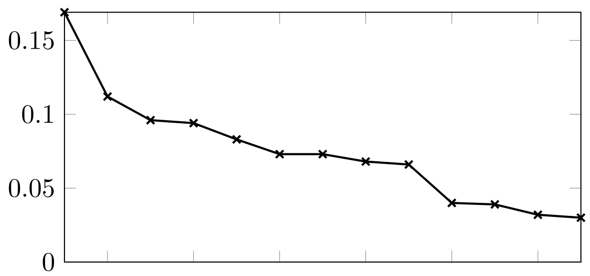


Figure 2: Relative density of communities in industrial bounded model checking instance in Figure 1a. For a community of size s, the community density is the fraction of $\binom{s}{2}$ pairs that are connected by an edge.

and only if u and v appear as literals together in a clause of F. The constraint graph of a formula is an important tool to understanding the structure of the formula, and how that structure might be exploited by SAT solvers.

The *modularity* of G (Newman and Girvan 2004) measures the strength of division into communities. Given a partition Γ of V into communities $\Gamma_1, \Gamma_2, \ldots$, the modularity of G with respect to Γ is the fraction of edges within a community minus the expected fraction if the edges were randomly distributed. Given a community $\Gamma_\ell \in \Gamma$, denote as $E(\Gamma_\ell) \subseteq E$ the set of edges within the community. The modularity of G with respect to Γ is

$$Q(G, \Gamma) = \sum_{\Gamma_\ell \in \Gamma} \left(\frac{|E(\Gamma_\ell)|}{|E|} - \left(\frac{\sum_{v \in \Gamma_\ell} \deg(v)}{2|E|} \right)^2 \right).$$

The modularity of G is $Q(G) = \max_\Gamma Q(G, \Gamma)$. For a community-structured SAT formula $F \in F(n, m, k, s, p)$, the expected modularity of the underlying constraint graph of F is bounded by $\mathrm{E}(Q(G)) \geq p - s/n$ (Giráldez-Cru and Levy 2016, Theorem 2).

The (1+1) EA is a basic randomized search heuristic that functions as a kind of degenerate case for more complicated evolutionary techniques. It maintains a population size of one, and produces a single offspring via a mutation operation in each generation. Selection is performed by comparing the offspring to its parent. The parent is replaced only if the offspring is at least as fit as the parent. The mutation operation changes the current individual in a minimal way. In the case of length-n binary strings, the state of each variable is negated with probability $1/n$. In this way, the (1+1) EA, illustrated in Algorithm 1 operates similarly to a simple local search hill-climber (such as GSAT), with the distinction that in each step, larger jumps in the search space are possible. The probability that mutation produces a jump of Hamming distance d falls off as a degree-d polynomial in n. The distinction between the (1+1) EA and a simple hill-climber is small, but important. Current running time bounds for local search-like heuristics on logarithmically dense formulas of the uniform model depend on the ability of the algorithm to change more than one variable at a time (Doerr, Neumann, and Sutton 2017).

Algorithm 1: (1+1) EA

```
Choose $x$ uniformly at random from $\{0,1\}^n$;
while stopping criterion not met do
    $y \leftarrow x$;
    foreach $i \in \{1, \ldots, n\}$ do
        With probability $1/n$, $y_i \leftarrow (1 - y_i)$;
    if $f(y) \geq f(x)$ then $x \leftarrow y$;
```

Search space of planted modular formulas

As is usual with evolving assignments to propositional satisfiability formulas, we consider each full assignment to n Boolean variables as strings over a binary alphabet. This allows us to identify each assignment with a fitness function $f \colon \{0,1\}^n \to \{0, 1, \ldots, m\}$ such that $f(x)$ counts the number of clauses satisfied by the assignment corresponding to the binary string x.

Similar to previous analyses on propositional satisfiability (Sutton and Neumann 2014; Doerr, Neumann, and Sutton 2017; Buzdalov and Doerr 2017), we will rigorously characterize the *fitness-distance correlation* in typical fitness landscapes induced by f together with the Hamming metric on $\{0,1\}^n$. Fitness-distance correlation was introduced decades ago as a measure of difficulty for evolutionary algorithms (Jones and Forrest 1995), and describes the relatedness of fitness values to the distance to an optimal solution. In the context of planted formulas, we prove that the fitness of a point is strongly correlated to the distance from the planted solution (although a search algorithm may discover a different satisfying assignment before reaching the planted solution).

In the community attachment model, it will be useful to partition the fitness function by contribution from different communities. We take $f_\ell(x)$ to be the count of localized clauses from community ℓ that are satisfied by x, and $f_{\text{sep}}(x)$ to be the count of separated clauses satisfied by x. Thus

$$f(x) = f_{\text{sep}}(x) + \sum_{\ell=1}^{t} f_\ell(x). \qquad (1)$$

Furthermore, when $\mathcal{I} \subseteq [t]$, we take $f_{\mathcal{I}}(x)$ to mean the contribution $\sum_{\ell \in \mathcal{I}} f_\ell(x)$ to f from the localized clauses in the communities indexed by $\mathcal{I}$.

In the following section, we introduce a number of definitions and preliminary lemmas that support the main result.

Preliminaries

We define the following distance notation, keeping in mind that $x^\star$ corresponds to a fixed but arbitrary planted assignment.

Definition 1. *For $x, y \in \{0,1\}^n$, we denote the Hamming distance of x and y as $d(x,y)$. For a community $\Gamma_\ell \subseteq [n]$, we denote*

$$d(x, y; \Gamma_\ell) = |\{i \in \Gamma_\ell : x_i \neq y_i\}|$$

to be the Hamming distance of x and y restricted to community Γ_ℓ. Finally, when working with a planted solution $x^\star$, *we often simply use $d(x)$ and $d(x;\Gamma_\ell)$ to denote $d(x,x^\star)$ and $d(x,x^\star;\Gamma_\ell)$, respectively.*

We require the following definition that characterizes binary strings that are in some sense not too far from the planted solution when restricted to any community.

Definition 2. *For a formula in $\mathcal{F}_{x^\star}(n,m,k,s,p)$, we say a point $x\in\{0,1\}^n$ is ϵ-balanced with respect to the formula if $d(x;\Gamma_\ell)\le s(1/2+\epsilon)$ for every $\ell\in\{1,2,\dots,n/s\}$.*

As expected, so long as s is not too small, most strings in $\{0,1\}^n$ are ϵ-balanced. Moreover, we will also implicitly use the straightforward fact that the ϵ-balance property is closed under steps toward the planted solution.

Lemma 1. *For every fixed $0<\epsilon<1/2$, if F is a formula in $\mathcal{F}_{x^\star}(n,m,k,s,p)$ with each $s=\omega(\log n)$, then a uniformly chosen string from $\{0,1\}^n$ is ϵ-balanced with respect to F with probability $1-e^{-\Omega(s)}$.*

Proof. Let $x\in\{0,1\}^n$ be chosen uniformly at random. Thus for any $i\in[n]$, $\Pr(x_i=x_i^\star)=1/2$. For an arbitrary partition Γ_ℓ, the probability that $d(x;\Gamma_\ell)>(1+2\epsilon)\frac{s}{2}$ is at most $\exp(-2s\epsilon^2/3)$. Applying a union bound over all $t<n$ communities, x violates ϵ-balance in at least one community with probability at most $\exp(-2s\epsilon^2/3+\ln t)=e^{-\Omega(s)}$. □

We also define the following expression to count the clauses in a particular clause set that have a different satisfiability state between two assignments.

Definition 3. *Let C be a set of clauses. For any $x,y\in\{0,1\}^n$, denote as $\Delta_{xy}(C)$ the number of clauses in C that are unsatisfied by x but satisfied by y.*

Finally, we will make use of the following simple proposition to count types of clauses in a formula.

Proposition 1. *Let $d\le k\le s$ be integers. Then*

$$\sum_{r=1}^{d} r\binom{d}{r}\binom{s-d}{k-r}=\frac{dk}{s}\binom{s}{k}$$

Proof. Since $r\binom{d}{r}=d\binom{d-1}{r-1}$, we can rewrite the sum on the LHS as

$$\begin{aligned} d\sum_{r=1}^{d}\binom{d-1}{r-1}\binom{s-d}{k-r} &= d\sum_{r=0}^{d-1}\binom{d-1}{r}\binom{s-d}{k-(r+1)}\\ &= d\binom{s-1}{k-1}, \end{aligned}$$

where we have applied the Chu-Vandermonde identity (Gould 1956). □

Fitness signal from localized clauses

For a particular community Γ_ℓ, denote as C_ℓ the set of all legal k-CNF clauses formed from Γ_ℓ that are satisfied by $x^\star$. For all $\ell\in\{1,2,\dots,t\}$, $|C_\ell|=(2^k-1)\binom{s}{k}$ since there are 2^k ways to negate variables chosen from any k-set of Γ_ℓ, but exactly one of them is unsatisfied by $x^\star$.

We count the potential contribution to the increase in fitness of any legal clause in C_ℓ for all the points that are closer to the planted solution.

Lemma 2. *Consider a community Γ_ℓ and let $x\in\{0,1\}^n$ be arbitrary. Let C_ℓ denote the set of localized clauses for Γ_ℓ that are satisfied by a planted assignment $x^\star$. Then*

$$\sum_{y:d(x)-d(y)=1}\Delta_{xy}(C_\ell)=\frac{d(x;\Gamma_\ell)k}{s}\binom{s}{k}.$$

Proof. Let $\mathcal{I}\subseteq\Gamma_\ell$ be the index set corresponding to the positions in x that differ from $x^\star$ in community Γ_ℓ. A clause in C_ℓ is unsatisfied by x but satisfied by some y with $d(x)-d(y)=1$ if (1) all its literals are false under x, and (2) it contains a literal l such that the variable associated with l appears in $\mathcal{I}$. If it contains $r\le|\mathcal{I}|$ literals with associated variables indexed by elements of $\mathcal{I}$, then that clause contributes exactly r times to the sum of $\Delta_{xy}(C_\ell)$ over the Hamming neighbors of x that are closer to $x^\star$.

Such a k-clause can be constructed by choosing the r variables from $\mathcal{I}$ and the remaining $k-r$ variables from $\Gamma_\ell\setminus\mathcal{I}$. The negation of the k variables in the clause is the unique negation pattern on the k chosen variables so that the clause is not satisfied by x. There are $\binom{|\mathcal{I}|}{r}\binom{s-|\mathcal{I}|}{k-r}$ ways to choose a k-set that corresponds to a clause that contributes r to the sum, and the total contribution of all such clauses to the sum is given by Proposition 1. Since $|\mathcal{I}|=d(x;\Gamma_\ell)$, the RHS of the claim follows from the proposition.

To ensure that we have not over- or under-counted, note that each of these clauses contain $r>0$ literals, each of which must be true under some y with $d(y)=d(x)-1$. For any one of these literals, there is a unique $i\in|\mathcal{I}|$ that corresponds to the variable that must be flipped to move from x to y. Since $d(x;\Gamma_\ell)-d(y;\Gamma_\ell)=1$, $y_i=x_i^\star$, and so that literal is also true under $x^\star$. It must therefore be the case that each of the clauses we have counted above are also satisfied by $x^\star$, and are hence all valid clauses in C_ℓ. □

We also need to count the potential contribution to the decrease in fitness of any legal clause in C_ℓ for all the points that are one step closer to the planted solution. This is captured by the following lemma.

Lemma 3. *Consider a community Γ_ℓ and let $x\in\{0,1\}^n$ be arbitrary. Let C_ℓ denote the localized clauses for Γ_ℓ that are satisfied by a planted assignment $x^\star$.Then*

$$\sum_{y:d(x)-d(y)=1}\Delta_{yx}(C_\ell)=d(x;\Gamma_\ell)\left(\binom{s-1}{k-1}-\binom{s-d(x;\Gamma_\ell)}{k-1}\right).$$

Proof. For each k-set of variables from Γ_ℓ, there is only one way to negate the variables to produce a unique clause (up to commutativity) that is not satisfied by an assignment y. There are $d(x;\Gamma_\ell)$ bitstrings y with $d(x)-d(y)=1$ that differ from x in exactly one position $i\in\Gamma_\ell$, and for each of these, $\binom{s-1}{k-1}$ ways to choose the remaining $k-1$ variables to appear in a clause. However, $\binom{s-d(x;\Gamma_\ell)}{k-1}$ of these clauses are not allowed because all k corresponding positions differ from $x^\star$, and hence such a clause would not be satisfied by $x^\star$, and is therefore not a legal clause of C_ℓ. Subtracting the count of these clauses yields the RHS of the claim. □

Lemma 4. *Let F be a planted modular formula from $\mathcal{F}_{x^\star}(n,m,k,s,p)$ with $k, \epsilon = \Theta(1)$, and let $\mathcal{I} \subseteq [t]$. With probability*

$$1 - \exp\left(-\Omega\left(\frac{mp}{s}\sum_{\ell\in\mathcal{I}}\pi_\ell d(x;\Gamma_\ell)\right)\right),$$

the following bound holds at every ϵ-balanced point x.

$$\sum_{y:d(x)-d(y)=1} f_{\mathcal{I}}(y) - f_{\mathcal{I}}(x) = \Theta\left(\frac{mp}{s}\sum_{\ell\in\mathcal{I}}\pi_\ell d(x,\Gamma_\ell)\right).$$

Proof. Fix x to be an arbitrary ϵ-balanced point. Let $N = \{y : d(x) - d(y) = 1\}$. Note that $x^\star$, x and N are fixed for the remainder of the proof. We define two sequences of m random variables over the probability space of randomly constructed formulas F from $\mathcal{F}_{x^\star}(n,m,k,s,p)$. Let α_i denote the i-th clause of F and let $\{Y_1, Y_2, \ldots, Y_m\}$ be the sequence

$$Y_i = \begin{cases} 0 & \text{if } \alpha_i \text{ is separated or true under } x, \\ |\{y \in N : \alpha_i \text{ is true under } y\}| & \text{otherwise.} \end{cases}$$

If α_i is localized, Y_i counts the total number of times that α_i switches from false to true in the set of Hamming neighbors of x that are strictly closer to the planted solution. Similarly, we define the sequence $\{Z_1, Z_2, \ldots, Z_m\}$ as

$$Z_i = \begin{cases} 0 & \text{if } \alpha_i \text{ is separated or false under } x, \\ |\{y \in N : \alpha_i \text{ is false under } y\}| & \text{otherwise.} \end{cases}$$

Since a clause can be false under only one configuration of its k variables, $Z_i \in \{0,1\}$. When α_i is localized, Z_i indicates the presence of an element of N for which α_i switches from true to false. As each of the m clauses is chosen independently, both $\{Y_i : i \in [m]\}$ and $\{Z_i : i \in [m]\}$ are sequences of independent random variables. For any $\ell \in \mathcal{I}$, we choose clause α_i from C_ℓ with probability $p\pi_\ell$. Furthermore, the probability that α_i is chosen from the set of clauses that change from false to true when moving from any x to y is exactly $\Delta_{xy}(C_\ell)/|C_\ell|$, and we calculate the expectation as follows.

$$\begin{aligned} \mathrm{E}(Y_i) &= p\sum_{\ell\in\mathcal{I}}\frac{\pi_\ell}{|C_\ell|}\sum_{y:d(x;\Gamma_\ell)-d(y;\Gamma_\ell)=1}\Delta_{xy}(C_\ell) \\ &= \frac{p}{(2^k-1)\binom{s}{k}}\sum_{\ell\in\mathcal{I}}\frac{\pi_\ell d(x;\Gamma_\ell)k}{s}\binom{s}{k}, \quad \text{by Lemma 2} \\ &= \frac{pk}{(2^k-1)s}\sum_{\ell\in\mathcal{I}}\pi_\ell d(x;\Gamma_\ell). \end{aligned}$$

Similarly,

$$\begin{aligned} \mathrm{E}(Z_i) &= p\sum_{\ell\in\mathcal{I}}\frac{\pi_\ell}{|C_\ell|}\sum_{y:d(x;\Gamma_\ell)-d(y;\Gamma_\ell)=1}\Delta_{yx}(C_\ell) \\ &= \tfrac{p}{(2^k-1)\binom{s}{k}}\textstyle\sum_{\ell\in\mathcal{I}}\pi_\ell d(x;\Gamma_\ell)\left(\binom{s-1}{k-1} - \binom{s-d(x;\Gamma_\ell)}{k-1}\right), \end{aligned}$$

which holds by Lemma 3. Furthermore, since x is supposed to be ϵ-balanced, $d(x;\Gamma_\ell) \leq s(1/2+\epsilon)$, and thus

$$\begin{aligned} &\leq \tfrac{p}{(2^k-1)\binom{s}{k}}\textstyle\sum_{\ell\in\mathcal{I}}\pi_\ell d(x;\Gamma_\ell)\left(\binom{s-1}{k-1} - \binom{s(1/2-\epsilon)}{k-1}\right) \\ &= \tfrac{p\sum_{\ell\in\mathcal{I}}\pi_\ell d(x;\Gamma_\ell)}{(2^k-1)}\left(\tfrac{k}{s} - \tfrac{k(1/2-\epsilon)^{k-1}}{s} + O\left(\tfrac{1}{s^2}\right)\right) \\ &= \tfrac{pk\sum_{\ell\in\mathcal{I}}\pi_\ell d(x;\Gamma_\ell)}{(2^k-1)s}(1 - (1/2-\epsilon)^{k-1} + O(1/s)). \end{aligned}$$

Note that the total fitness change in localized clauses indexed by $\mathcal{I}$ moving from x to all $y \in N$ is equivalent to the difference in the sums of the random variables that count localized clauses that switch state between x and all $y \in N$. In other words,

$$\sum_{y:d(x)-d(y)=1} f_{\mathcal{I}}(y) - f_{\mathcal{I}}(x) = \sum_{i=1}^{m}(Y_i - Z_i).$$

As $0 \leq Y_i \leq k$ and $0 \leq Z_i \leq 1$, setting $Y = \sum_{i=1}^m Y_i/k$ and $Z = \sum_{i=1}^m Z_i$, for any arbitrary constant $0 < \delta < 1$ the probability $Y \in [(1-\delta)\mathrm{E}(Y/k), (1+\delta)\mathrm{E}(Y)]$ and $Z \in [0, (1+\delta)\mathrm{E}(Z)]$ is bounded as claimed by applying a multiplicative Chernoff bound, completing the proof. □

Efficiently solving constant-density formulas

The result of Lemma 4 estimates the probability that a small local mutation can detect a move in the direction of the planted solution. In the original community attachment model, $\pi_\ell = 1/t$, that is, a localized clause is chosen uniformly over all communities. We consider a slight generalization in which communities are no longer required to be uniform in density. This generalization is arguably more realistic, as it more closely matches the variability in community density observed in real-world data (see Figure 2). For any small constant $0 < \varepsilon < 1$, we partition the communities so that a nonempty set $\mathcal{I} \subseteq [t]$ of communities are *dense* in the sense that $\pi_\ell = \Omega(1/t^{1-\varepsilon})$ for each $\ell \in \mathcal{I}$ and the remaining are *sparse* so that $\pi_\ell = O(1/t^{1+\varepsilon})$ for $\ell \in [t] \setminus \mathcal{I}$.

Theorem 1. *Let $0 < \varepsilon < 1$ be any small constant. Let $s \in \omega(\log n) \cap O(n^{\varepsilon/(2\varepsilon+2)})$, set $p = 1 - O(1/n^{1+\varepsilon})$ and $k = \Theta(1)$. Let $\emptyset \subsetneq \mathcal{I} \subseteq [t]$ with*

$$\pi_\ell = \begin{cases}\Omega\left(\frac{1}{t^{1-\varepsilon}}\right) & \text{for } \ell \in \mathcal{I}, \\ O\left(\frac{1}{t^{1+\varepsilon}}\right) & \text{for } \ell \in [t]\setminus\mathcal{I}.\end{cases}$$

For constraint densities $m/n \geq c$ where c is a sufficiently large constant, all but a vanishing fraction of planted modular formulas F in $\mathcal{F}_{x^\star}(n,m,k,s,p)$ are solved in polynomial expected time by the (1+1) EA.

Proof. We first argue that all but a fast-vanishing fraction of formulas in $\mathcal{F}_{x^\star}(n,m,k,s,p)$ have a good drift condition on $f_{\mathcal{I}}$. Let $x \in \{0,1\}^n$ be an arbitrary ϵ-balanced string. Define

$$N = \{y : d(x) - d(y) = 1 \land x_i \neq y_i \implies \exists \ell \in \mathcal{I}, i \in \Gamma_\ell\}$$

to be the set of $\sum_{\ell\in\mathcal{I}} d(x;\Gamma_\ell)$ neighbors that are closer to $x^\star$ in some dense community indexed by $\mathcal{I}$. Applying Lemma 4, the bound

$$\sum_{y\in N} f_{\mathcal{I}}(y) - f_{\mathcal{I}}(x) = \Theta\left(\frac{m}{st^{1-\varepsilon}}\sum_{\ell\in\mathcal{I}} d(x,\Gamma_\ell)\right)$$

holds with probability

$$1 - \exp\left(-\Omega\left(\frac{m}{st^{1-\varepsilon}} \sum_{\ell \in \mathcal{I}} d(x; \Gamma_\ell)\right)\right).$$

Since $f_{\mathcal{I}}$ is independent of any bit not in $\bigcup_{\ell \in \mathcal{I}} \Gamma_\ell$, applying a union bound over all ϵ-bounded length-$|\mathcal{I}|t$ substrings indexed by $\bigcup_{\ell \in \mathcal{I}} \Gamma_\ell$ lying at distance d from the substring of $x^\star$ indexed by $\bigcup_{\ell \in \mathcal{I}} \Gamma_\ell$, we find this strong drift condition holds with probability at least

$$1 - \sum_{d=1}^{n} \binom{|\mathcal{I}|t}{d} \exp\left(-\Omega(dn^{\varepsilon^2})\right) \geq 1 - \exp\left(-\Omega(n^{\varepsilon^2})\right).$$

Here we have used the fact that $m/(st^{1-\varepsilon}) = \Omega(n^{\varepsilon^2})$.

We now show that the fitness contribution from other clauses is small. The expected number of separated clauses in the entire formula is bounded as $cn(1-p) = O(1/n^{\varepsilon})$ which exceeds a constant only with probability $o(1)$ by Markov's inequality. Therefore, with probability $1 - o(1)$, at every ϵ-balanced point,

$$\begin{aligned}\sum_{y \in N} f(y) - f(x) &\geq \sum_{y \in N} f_{\mathcal{I}}(y) - f_{\mathcal{I}}(x) - O(1) \\ &= \Omega\left(n^{\varepsilon^2} \sum_{\ell \in \mathcal{I}} d(x, \Gamma_\ell)\right).\end{aligned}$$

Let $d = \sum_{\ell \in \mathcal{I}} d(x; \Gamma_\ell)$. Recall that N is the set of all Hamming neighbors of x that are closer to $x^\star$ by some variable in a dense community (those indexed by $\mathcal{I}$). A simple proof by induction on distance in dense communities from the planted solution (see Lemma 6 of (Doerr, Neumann, and Sutton 2017)) yields that $f_{\mathcal{I}}(x) + f_{\text{sep}}(x) = O(d)$ since $m/n = \Theta(1)$. As any Hamming neighbor of x is generated by standard mutation with probability $\left(1 - \frac{1}{n}\right)^{n-1} \frac{1}{n} \geq e^{-1}/n$, we can bound the drift of $f_{\mathcal{I}} + f_{\text{sep}}$ at every ϵ-balanced point x by

$$\begin{aligned}\frac{1}{en} \sum_{y \in N} \max\{0, f(y) - f(x)\} &= \Omega\left(\frac{dn^{\varepsilon^2}}{n}\right) \\ &= \Omega\left(\frac{f_{\mathcal{I}}(x) + f_{\text{sep}}(x)}{n^{1-\varepsilon^2}}\right).\end{aligned} \tag{2}$$

As long as the the (1+1) EA never generates a point that is not ϵ-balanced with respect to the dense communities, there is sufficient drift toward the planted solution in the length-$|\mathcal{I}|t$ substring induced by the dense communities. The initial point is drawn uniformly at random, so Lemma 1 guarantees an ϵ-balanced string with probability at least $1 - e^{-\Omega(n^{\epsilon})}$. Finally, as long the current point x of the (1+1) EA is ϵ-balanced, the probability of reaching a non-ϵ-balanced point within any polynomial number of iterations is superpolynomially close to zero by using Lemma 4 together with the negative drift theorem (Oliveto and Witt 2011; 2012) at every dense community. Applying the multiplicative drift theorem (Doerr, Johannsen, and Winzen 2012), the dense communities are solved in expected time $O(n^{1-\varepsilon^2} \log n)$.

It remains to bound the time until the remaining variables are solved. Fix $\ell \in [t] \setminus \mathcal{I}$ to be the index to an arbitrary sparse community, and let $\{X_i : 1 \leq i \leq m\}$ be the indicator random variable for localized clauses from Γ_ℓ. The expected number of localized clauses in the community is $\mathrm{E}\left(\sum_{i=1}^{m} X_i\right) = cn\pi_\ell = O\left(\frac{1}{n^{\varepsilon/2}}\right)$, since for sparse communities we have $\pi_\ell = O(t^{-(1+\epsilon)})$ and we have assumed the community-size bound of $s = O(n^{\varepsilon/(2\varepsilon+2)})$.

Finally, we apply a standard Chernoff bound to the sum of indicator random variables to ensure that the sum only exceeds a constant $b > 1$ with probability $O(1/n^{b\varepsilon})$. Thus each of the $O(t)$ sparse communities contribute at most b localized clauses to f with $O(1/n^{\varepsilon})$ probability. Since we have assumed there are only a constant number of separated clauses in total, the total number of clauses associated with bits in Γ_ℓ is a constant, and each of the remaining community substrings can be optimized in $O(tn^{O(1)}) = n^{O(1)}$ time, concluding the proof. □

Experiment Results

In order to characterize the leading constants and demonstrate the tightness of the bound, we perform a number of numerical experiments to measure the run time of the (1+1) EA on the community attachment model. Since our proofs only require the constraint density to be a sufficiently large but unspecified constant, we generate run length distribution (RLD) curves of the (1+1) EA as a function of constraint density to observe its behavior. In Figure 3, we fix $n = 1000$, $s = 100$ and $p = 3/4$ and vary m to plot the empirical RLD curves.

On the standard uniform planted model there is a critical constant density near which the performance of the (1+1) EA degrades significantly (Doerr, Neumann, and Sutton 2017). There is likely to be a similar effect in the community attachment model, and we conjecture that the performance degradation as density decreases is an expression of this effect. According to the RLD curves of Figure 3, the density of $m/n = 5e$ is sufficiently large to escape this deterioration. We therefore choose this constant to be large enough of a density for our run time experiments.

For each $s = \{100, 110, 120, \ldots, 1000\}$, we generate 10 modular formulas using the community attachment model with $n = s^{3/2}$ and $m/n = 5e$. On each formula, we measure the run time of the (1+1) EA for 10 trials. To establish a segmentation into dense and sparse communities, we identify $\lfloor t^{(1-\varepsilon)} \rfloor$ communities as *dense* and choose localized clauses uniformly from dense communities with probability $1 - t^{-\varepsilon}$. The remaining communities are chosen uniformly with probability $t^{-\varepsilon}$. Thus a localized clause from a particular dense community is generated with probability $(1 - t^{-\varepsilon})/\lfloor t^{(1-\varepsilon)} \rfloor = \Omega(1/t^{(1-\varepsilon)})$ and from a particular sparse community with probability $t^{-\varepsilon}/(t - \lfloor t^{(1-\varepsilon)} \rfloor) = O(1/t^{1+\varepsilon})$ as required by the proof of Theorem 1. We conjecture, however, that the asymptotic behavior of the (1+1) EA does not depend strongly on this condition. Nor do we believe that the requirement for vanishing separated clauses is necessary.

In Figure 4, we plot the median run time divided by $n \ln n$

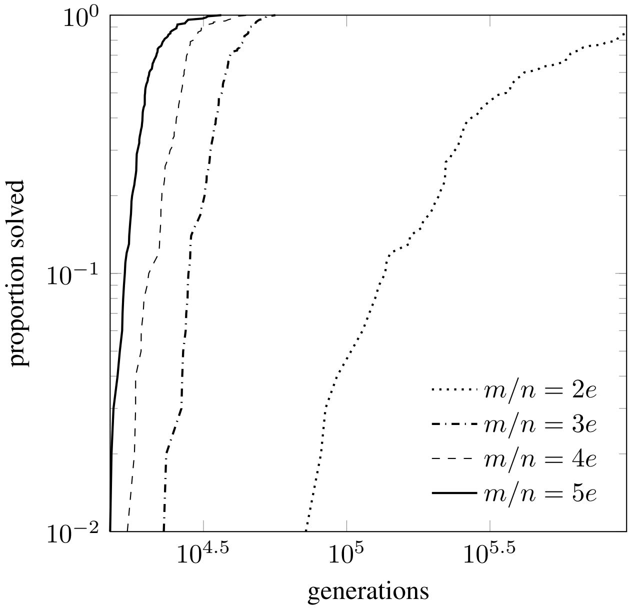


Figure 3: Empirical run length distributions on community attachment model, controlling for density. The curves are created by 100 runs of the (1+1) EA at each constraint density value $m/n \in \{2e, 3e, 4e, 5e\}$. Each formula contains $n = 1000$ variables, community size $s = 100$, and localized clause probability $p = 3/4$.

measured in the experiments for $p \in \{3/4, 1/4\}$ and $\varepsilon = 1/5$. We also compare the run time of the (1+1) EA solving formulas generated with the community attachment model over uniform communities.

Empirically, the median run time converges to roughly $3.1 \cdot n \ln n$, thus our time bound for solving the dense communities appears to be closer to the truth, even when the count of separated clauses is non-vanishing. We also find support for our conjecture that the uniformity of communities does not change the asymptotic behavior dramatically, and overall, the running time seems to be largely invariant to localized clause probability.

Conclusions

The uniform random SAT model has been studied intensely for decades. Recently, several works have introduced non-uniform models that produce formulas exhibiting structural features that correspond better to real-world industrial instances. In this paper we push the field of run time analysis of randomized search heuristics toward these non-uniform models by conducting a rigorous analysis of the running time of the (1+1) EA on the recently proposed community attachment random SAT model, which has tunable modularity. We prove that, as long as the the community size is polynomially smaller than the number of variables, the EA can be efficient down to constant constraint densities. This can be contrasted to the uniform random model, on which the best known results require a density of $\Omega(\log n)$. Our analysis leverages the fact that in the community-structured formulas the fitness signal uncovered by mutations concentrates around the frequency of dense localized clauses, rather than at the overall constraint density. Nevertheless, we conjecture that the asymptotic behavior of the EA is similar on constant-density uniform formulas.

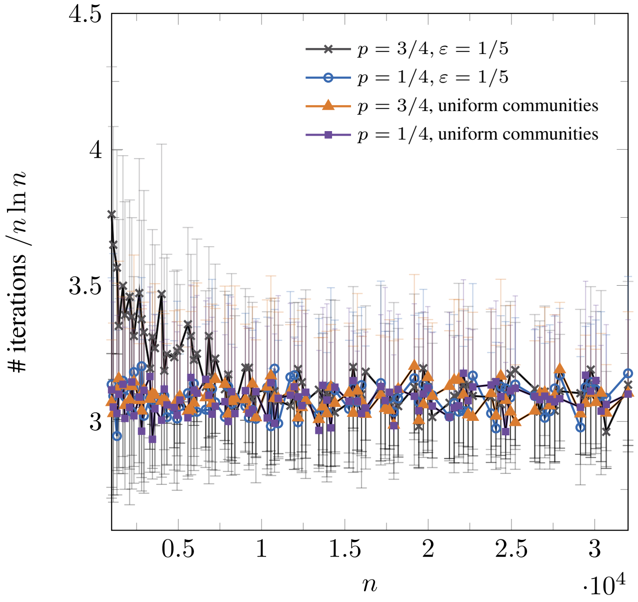


Figure 4: Median run time of the (1+1) EA divided by $n \ln n$ as a function of n on formulas sampled from the community attachment model, $p \in \{3/4, 1/4\}$, and $m = 5en$. Both non-uniform communities ($\varepsilon = 1/5$) and uniform communities are plotted. The error bars denote the interquartile range. The statistics are taken from 10 runs each on 10 random formulas generated for each value of $n = s^{3/2}$ varying community size s from 100 to 1000.

References

Ansótegui, C.; Bonet, M. L.; Giráldez-Cru, J.; and Levy, J. 2014. The Fractal Dimension of SAT Formulas. In *Proceedings of the Seventh International Joint Conference on Automated Reasoning (IJCAR)*, 107–121. Springer.

Ansótegui, C.; Bonet, M. L.; and Levy, J. 2009. On the structure of industrial SAT instances. In *Proceedings of the Fifteenth Conference on Constraint Programming (CP)*, 127–141.

Bastian, M.; Heymann, S.; and Jacomy, M. 2009. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media (ICWSM).*

Buzdalov, M., and Doerr, B. 2017. Runtime analysis of the (1 + (λ, λ)) genetic algorithm on random satisfiable 3-CNF formulas. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 1343–1350. ACM.

Doerr, B.; Johannsen, D.; and Winzen, C. 2012. Multiplicative drift analysis. *Algorithmica* 64:673–697.

Doerr, B.; Neumann, F.; and Sutton, A. M. 2017. Time complexity analysis of evolutionary algorithms on random satisfiable k-CNF formulas. *Algorithmica* 78(2):561–586.

Giráldez-Cru, J., and Levy, J. 2016. Generating SAT instances with community structure. *Artificial Intelligence* 238:119 – 134.

Giráldez-Cru, J., and Levy, J. 2017. Locality in random SAT instances. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, 638–644.

Gould, H. W. 1956. Some generalizations of Vandermonde's convolution. *The American Mathematical Monthly* 63(2):84–91.

Jones, T., and Forrest, S. 1995. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA)*, 184–192. Morgan Kaufmann.

Kautz, H. A.; Ruan, Y.; Achlioptas, D.; Gomes, C. P.; Selman, B.; and Stickel, M. E. 2001. Balance and filtering in structured satisfiable problems (preliminary report). *Electronic Notes in Discrete Mathematics* 9:2–18.

Krivelevich, M., and Vilenchik, D. 2006. Solving random satisfiable 3CNF formulas in expected polynomial time. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 454–463. ACM Press.

Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69:026113.

Oliveto, P. S., and Witt, C. 2011. Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica* 59(3):369–386.

Oliveto, P. S., and Witt, C. 2012. Erratum: Simplified drift analysis for proving lower bounds in evolutionary computation. `arXiv:1211.7184 [cs.NE]`.

Schöning, U. 1999. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *Proceedings of the Fortieth Annual Symposium on Foundations of Computer Science (FOCS)*, 410–414. IEEE Computer Society.

Selman, B.; Kautz, H. A.; and Cohen, B. 1994. Noise strategies for improving local search. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI)*, 337–343.

Sutton, A. M., and Neumann, F. 2014. Runtime analysis of evolutionary algorithms on randomly constructed high-density satisfiable 3-CNF formulas. In *Proceedings of the Thirteenth International Conference on Parallel Problem Solving from Nature (PPSN)*, volume 8672 of *Lecture Notes in Computer Science*, 942–951. Springer.